Patents Office
Government Buildings
Hebron Road
Kilkenny

I HEREBY CERTIFY that annexed hereto is a true copy of documents filed in connection with the following patent application:

| | |
|---|---|
| Application No. | 2001/0744 |
| Date of Filing | 3 August 2001 |
| Applicant | SIVTECH LIMITED an Irish company of International House, Tara Street, Dublin 2, Ireland |

Dated this 27 day of January 2004.

An officer authorised by the
Controller of Patents, Designs and Trademarks.

FORM NO: 1

010744

# REQUEST FOR THE GRANT OF A PATENT

## PATENTS ACT, 1992

The Applicant(s) named herein hereby request(s)

X the grant of a patent under Part II of the Act

______ the grant of a short-term patent under Part III of the Act

on the basis of the information furnished hereunder.

1. Applicant(s)

Name Sivtech Limited
Address International House
Tara Street
Dublin 2
Ireland

Description/Nationality

An Irish company

2. Title of Invention

"A data quality system"

3. Declaration of Priority on basis of previously filed application(s) for same invention (Sections 25 & 26)

| Previous filing date | Country in or for which filed | Filing No. |
|---|---|---|

4. Identification of Inventor(s)
Name(s) of person(s) believed by Applicants(s) to be the inventor(s)

MORONEY, Garry
an Irish citizen of 47 Beaverbrook, Donabate, County Dublin, Ireland

CAULFIELD, Brian
an Irish citizen of 100 Iona Road, Glasnevin, Dublin 9, Ireland

PEARCE, Ronan
an Irish citizen of 8 Salzburg, Ardilea, Dublin 14, Ireland

CUNNINGHAM, Padraig
an Irish citizen of 50 Avondale Lawn, Blackrock, County Dublin, Ireland

DELANEY, Sarah-Jane
an Irish citizen of 50 Avondale Lawn, Blackrock, County Dublin, Ireland

RAMSEY, Gary
an Irish citizen of 25 Hillcourt Road, Glenageary, County Dublin, Ireland

5. Statement of right to be granted a patent (Section 17(2) (b)

The Applicant derives the rights to the Invention by virtue of a Deed of Assignment dated May 15, 2001

6. Items accompanying this Request – tick as appropriate

| | | |
|---|---|---|
| (i) | X | prescribed filing fee (£100.00) |
| (ii) | X | specification containing a description and claims |
| | ____ | specification containing a description only |
| | X | Drawings referred to in description or claims |
| (iii) | ____ | An abstract |
| (iv) | ____ | Copy of previous application (s) whose priority is claimed |
| (v) | ____ | Translation of previous application whose priority is claimed |
| (vi) | X | Authorisation of Agent (this may be given at 8 below if this Request is signed by the Applicant (s) |

7. Divisional Application (s)

The following information is applicable to the present application which is made under Section 24 –

Earlier Application No: ..................

Filing Date: ..................

8. Agent

The following is authorised to act as agent in all proceedings connected with the obtaining of a patent to which this request relates and in relation to any patent granted -

| Name | Address |
|---|---|
| John A. O'Brien & Associates | The address recorded for the time being in the Register of Patent Agents, and currently Third Floor, Duncairn House, 14 Carysfort Avenue, Blackrock, Co. Dublin, Ireland. |

9. Address for Service (if different from that at 8)

As above

Signed ____________ JOHN A. O'BRIEN & ASSOCIATES

Date August 3, 2001

"A data quality system"

Introduction

The invention relates to a data quality system.

Data quality is p important for companies maintaining large volumes of information in the form of structured data. It is becoming an increasingly critical issue for companies with very large numbers of customers (e.g. banks, utilities airlines)Many such companies have already, or are about to, implement customer relationship management (CRM) systems to improve their business development. Effective operation of CRM systems involves drawing data from a range of operational systems and aggregating it on a customer-by-customer basis. This involves a large degree of data matching based on criteria such as customer identification details. Such matching and associated operations are often ineffective because of bad quality data, thus undermining the CRM operations, for example. In more detail, data matching difficulties arise from (a) the multitude of different ways in which two equivalent sets of data can differ, and (b) the very large volumes of data generally involved. This means that carrying out the task manually is impossible or hugely costly and defining a finite set of basic matching rules to automate the process is extremely difficult. As organisations collect more data from more sources and attempt to use this data efficiently and effectively they are encountering this problem more frequently and the negative impact is growing.

It is therefore an objective of the invention to provide a data quality system to improve data quality.

Statements of Invention

According to the invention, there is provided a data quality system for matching input data across data records where the type and volume of input data is highly flexible, the system comprising:-

means for carrying out pre-processing routines on the input data to remove noise or reformat the data, and

means for matching record pairs based on measuring similarity of selected field pairs within the record.

In one embodiment, the matching means comprises means for extracting a similarity vector by generating a similarity score for each pair of fields in records being matched, a set of scores for a pair of records being a vector.

In another embodiment, the vector extraction means comprises means for executing string matching routines on pre-selected fields of the records.

In a further embodiment, wherein the matching means comprises record scoring means for converting the vector into a single similarity score representing overall similarity of two records.

In one embodiment, the record scoring means comprises means for executing rule-based routines using weights applied to fields according to the extent to which each field is indicative of record matching.

In another embodiment, the record scoring means comprises means for computing scores using AI (artificial intelligence) techniques to deduce from examples given by the user the optimum algorithm for computing the score from the vector.

In a further embodiment, the AI technology used is Cased Based Reasoning (CBR).

In one embodiment, AI technique used is Neural Nets.

In another embodiment, the pre-processing means comprises a standardisation module comprises means for transforming each data field into a number of data fields each of which is a variation of the original.

In a further embodiment, the standardisation module comprises means for splitting a data field into multiple field elements, converting the field elements to a different format, removing noise characters, and replacing elements with equivalent elements selected from an equivalence table.

In another embodiment, the pre-processing means comprises a grouping module comprises means for grouping records according to features to ensure that all actual matches of a record are within a group.

In a further embodiment, the grouping module comprises means for applying a key letter process for grouping.

In one embodiment, the system further comprises a configuration manager comprises means for applying configurable settings for the pre-processing means and the matching means.

Detailed Description of the Invention

The invention will be more clearly understood from the following description of some embodiments thereof, given by way of example only with reference to Fig. 1, which is a block diagram illustrating a data quality system of the invention.

Referring to Fig. 1, a data quality system 1 comprises a user interface 2 linked with a configuration manager 3 and a tuning manager 4. A data input adapter 5 directs input data to a pipeline 6 which executes to perform data matching in a high-speed and accurate manner.. The pipeline 6 comprises:

a pre-processor 7 having a standardisation module 8 and a grouping module 9,

a matching system 11 comprising a similarity vector extraction module 12 and a record scoring module 13.

The output of the pipeline 6 is fed to an output datafile 15.

The system 1 operates to match equivalent but non-identical information. This matching enables records to be amended to improve data quality.

The system 1 ("engine") processes one or multiple datasets to create an output data file containing a list of all possible matching record pairs and a *similarity score.* Depending on the needs of the user the engine can then automatically mark certain record pairs above a specified score as definite matches and below a specified score as non-matches. Record pairs with scores between these two thresholds may be sent to a user interface for manual verification.

There are a number of discrete activities within the matching process. These can be grouped into two separate phases: – **pre-processing** and **matching**

**Pre-processing**

In the pre-processing phase all data records are read sequentially from the data input adapters. Firstly each record is fed to the standardisation module 8 where a range of different routines are applied to generate an output record which can be matched

more effectively with other records. Each record is then fed through the **grouping module 9**. In this process labels are attached to the record to enable it to be easily and quickly grouped with other similar records. This makes the following matching process more efficient as it eliminates the need to compare records which are definitely non matches. Following the grouping process the output record (transformed and labelled) is written to the pre-processed datafile.

**Matching**

In the matching phase, each record is read in sequence from the pre-processed dataset 10. It is then compared to each similar record in the dataset – i.e. records within the same group. The comparison process involves:

1. **Similarity Vector Extraction**: This involves comparing individual fields within a record pair using matching algorithms to generate a similarity score for each pair of fields. Data element scoring is carried out on a number of field pairs within the record pair to generate a set of similarity scores called a similarity vector.
2. **Data record Scoring**: Once a similarity vector has been produced for a record pair by a series of data element scoring processes, the data record scoring process converts the vector into a single similarity score. This score represents the overall similarity of the two records.

The pair of output records is then written to the output datafile along with the similarity score. The matching phase then continues with the next pair of possible matching pairs.

To achieve high accuracy matching, the setup of the modules is highly specific to the structure and format of the dataset(s) being processed. A key advantage of the engine is built-in intelligence and flexibility which allow easy configuration of optimum

setup for each of the modules. Initial setup of the four processing modules is managed by the **Configuration Manager 3** and the **Tuning Manager 4**.

Standardisation ("Transformation")

Module 8

**Objective**

The aim of the transformation process is to remove many of the common sources of matching difficulty while ensuring that good data is not destroyed in the process. This is done by transforming the individual elements of a record into a range of different formats which will aid the matching process. Each data field in a record is transformed into a number of new data fields each of which is a variation of the original.

**Process**

Each data record is read in turn from the adaptor 5. Each field within a record is processed by applying a number of predefined transformation routines to the field. Each transformation routine produces a new output data field. Thus, an output record is produced containing a number of data fields for each field in the input record.

Field transformation routines include:

- Splitting a data field into multiple fields, for example Splitting street address into number, name and identifier.
- Converting field elements to other format using conversion routines, for example:
  - Converting to uppercase.
  - Converting to phonetic code (Soundex).
  - Convert to abbreviated version.

- Convert to standardised format (e.g. international telephone codes).
- Convert to business-specific version.

- Removal of characters from within data field, for example:
  - Removal of spaces between specified elements.
  - Removal of specified symbols from between specified elements (e.g. punctuation marks / hyphens).

- Replacement of element with an equivalent element selected from an equivalence table, for example:
  - Replacement of nickname / shortened name with rootname.
  - Replacement of Irish/foreign language place or person name with English equivalent.
  - Replacement of standard abbreviations with root term (st. to street, rd. to road etc.).
  - Replacement of company name with standardised version of name.

**Module Design**

The transformation module 8 is capable of a carrying out a user-defined number of transforms such as those above to each input data field and generating a user-defined number of output fields for each input field. The transforms required for each field type may be configured by:

- Selecting from a menu of default transformation configurations (set of routines) predefined by SiVTech for use with a particular field type of a particular structure/format/quality level.
- Developing their own configurations for each data field / element from a menu of transformations such as those above.

- Developing their own configurations for each data field / element using bespoke transformations input by the user – probably combined with some predefined transformations.

In batch matching projects the transformation process will be carried out on the whole database before any matching is done. A new data file of transformed elements is then created for use in the matching process. This saves time by ensuring that the minimum number of transformations N are carried out (where N = number of records in the database) rather than the potential maximum number of transformations NxN. However in realtime search and match type implementations the transformation process will be carried out directly before the matching process for each record.

**Transformation Example**

Input Record:

| Firstname | Surname | Address1 | Address2 | Address3 | DOB | Telephone |
|---|---|---|---|---|---|---|
| John | O'Brien | 3 Oak Rd. | Douglas | Co. Cork | 20/4/66 | 021-234678 |

Output Record

| FN_stan | FN_Soundex | FN_Root | SN_stan | SN_Soundex | SN_root | A1_Num |
|---|---|---|---|---|---|---|
| John | Jon | Jonathon | OBrien | O-165 | Brien | 3 |
| **A1_text** | **A1_text_soundex** | **A1_st** | **A2_text** | **A2_str_soundex** | **A3_st** | **A3_text** |
| Oak | O-200 | Road | Douglas | Duglass | County | Cork |
| **DOB_Eur** | **DOB_US** | **Telephone** | **Tel_local** | | | |
| 20041966 | 04201966 | 35321234678 | 234678 | | | |

<u>Grouping Module 9</u>

**Objective**

The aim of the data record grouping process is to significantly speed up the matching step by reducing the number of record pairs which go through the set of complex match scoring routines. This is done by grouping records which have certain similar features – only records within the same group are then compared in the matching phase. (This greatly reduces the number of matching steps required from NxN to GxHxH where G is the number of groups and H is the number of elements per group).

The challenge is to ensure that all actual matches of any record are contained within the same group. The group labelling process must be kept simple so that minimal processing time is required to identify elements in the same group. In addition, to have a real impact on efficiency the groups must be substantially smaller than the full dataset (at least 10 times?)

**Process**

After the transformation process is performed on an individual data record a further set of predefined routines is applied to the certain fields of the record. These routines extract certain features from the data fields. These features are included in a small number (2-4) extra data fields appended to the output record. These labels allow the record to be grouped with other similar records.

The key attributes of the labels are:

- Must be very high probability (99.999%) that all matching records have some or all of the same labels.
- Labels must be easily extracted from the data fields.
- Labels must be impervious to the range data errors which will have not been corrected by the transformation process e.g.
  - Spelling errors

- Typing errors
- Different naming conventions
- Mixed fields

The grouping process is designed as a high speed filtering process to significantly reduce the amount of matches required rather than as a substitute for the matching process.

As such, in order to keep the labelling / grouping process simple but ensure that no matches are missed, each groups will be extremely large and the vast majority of records within a group will not match.

An example of the type of routine used in the labelling process is the **Keyletter** routine. The **keyletter** is defined as the most important matching letter in the field - generally first letter of main token – J for John, B for oBrien, O for Oak, D for Douglas, C for Cork. For example the label fields may then contain : first letters of firstname, surname, address1 and address 2.

The grouping criteria may then be set to: X(2 to 4) number of common labels. Matching would then only be carried out on records whose label fields contained 2 or more of the same letters. The **Keyletter** may also be derived from the soundex fields.

**Module Design**

In many cases **keyletter** may not be the appropriate labelling routine. The grouping module must have the flexibility to allow the user to define a number of bespoke labelling routines appropriate to the dataset (for example – if a particular data element within a dataset has a particularly high confidence level, grouping may be focused largely on this). He may do this by:

a. selecting a default grouping configuration predefined for this type of dataset,
b. firstly selecting the most appropriate fields, secondly selecting the appropriate labelling routines from a menu, thirdly defining the grouping criteria for the labels, or
c. as above but inputting customised labelling routines

**Example**

Input Record:

| Firstname | Surname | Address1 | Address2 | Address3 | DOB | Telephone |
|---|---|---|---|---|---|---|
| John | O'Brien | 3 Oak Rd. | Douglas | Co. Cork | 20/4/66 | 021-234678 |

Output Record

| FN_stan | FN_Soundex | FN_Root | SN_stan | SN_Soundex | SN_root | A1_Num |
|---|---|---|---|---|---|---|
| John | Jon | Jonathon | OBrien | O-165 | Brien | 3 |
| **A1_text** | **A1_text_soundex** | **A1_st** | **A2_text** | **A2_str_soundex** | **A3_st** | **A3_text** |
| Oak | O-200 | Road | Douglas | Duglass | County | Cork |
| DOB_Eur | DOB_US | Telephone | Tel_local | | | |
| 20041966 | 04201966 | 35321234678 | 234678 | | | |

Output Record Grouping Labels

| FN_keyletter | SN_keyletter | A1_keyletter | A2_keyletter | A3_keyletter |
|---|---|---|---|---|
| J | B | O | D | C |

Similarity Vector Extraction Module 12

**Objective**

Each data field within a record is compared with one or more fields from the other record. The challenge here is to ensure that equivalent data elements are matched

using an appropriate matching routine even if the elements are not stored in equivalent fields.

**Process**

Each pair of records is read into the vector extraction module from the preprocessed datafile. This module firstly marks the data fields from each record which should be compared to each other. It then carries out the comparison using one of a range of different string matching routines. String matching routines are algorithms designed to accurately estimate the "similarity" of two data elements. Depending on the type / format of the data elements being compared, different matching routines are required. E.g. for a normal word the "edit distance" routine which measures how many edits required to change one element to the other is a suitable comparison routine. However for an integer it is more appropriate to use a routine which takes into account the difference between each individual digit and the different importance level of the various digits (i.e in number 684 the 6 is more important than the 8 which is more important than the 4)
Examples of matching routines are:

- Edit distance
- Hamming Distance
- Dyce
- Least Common substring

The output of the matching routine is a score between 0 and 1 where 1 indicates an identical match and 0 indicates a definite nonmatch.

The output of the data field scoring module is a set of *similarity scores* one for each of the datafield pairs compared. This set of scores is called a *similarity vector.*

**Module Design**

The module is designed to allow the user to select the data fields within the dataset/(s) to be used in the matching process, to select which fields are to be matched with which and to define the matching routine used for each comparison.

The user configures the process by:

- selecting from a menu of default configurations suitable for the dataset(s),
- manually selecting the data fields to be compared and selecting the appropriate matching routine from a menu of predefined routines, and
- Manually creating customised matching routines to suit particular data field types.

**Example**

Input Record 1

| **FN_stan** | **FN_Soundex** | **FN_Root** | **SN_stan** | **SN_Soundex** | **SN_root** | **A1_Num** |
|---|---|---|---|---|---|---|
| John | J-500 | Jonathon | OBrien | O-165 | Brien | 3 |
| **A1_text** | **A1_text_soundex** | **A1_st** | **A2_text** | **A2_str_soundex** | **A3_st** | **A3_text** |
| Oak | O-200 | Road | Douglas | D242 | County | Cork |
| DOB_Eur | DOB_US | Telephone | Tel_local | | | |
| 20041966 | 04201966 | 35321234678 | 234678 | | | |

Input Record 2

| **FN_stan** | **FN_Soundex** | **FN_Root** | **SN_stan** | **SN_Soundex** | **SN_root** | **A1_Num** |
|---|---|---|---|---|---|---|
| Jon | J-500 | Jonathon | Bryan | B-650 | Brien | - |
| **A1_text** | **A1_text_soundex** | **A1_st** | **A2_text** | **A2_text_sdx** | **A2_st** | **A3_text** |
| Oakdale | O-234 | Close | Oake | 0-230 | Road | Duglass |
| A4_st | A4_text | A4_text_sdx | DOB_Eur | DOB_US | Telephone | Tel_local |
| County | Cork | C-620 | 02041968 | 04021968 | | |

Output Similarity Vector

| | | | | | | |
|---|---|---|---|---|---|---|
| **FN_stan** | **FN_Root** | **SN_stan** | **SN_root** | **A1_Num** | **A1_text** | **A1_st** |
| .7 | 1 | .5 | 1 | .5 | .5 | 0 |
| **A2_text** | **A2_st** | **A3_text** | A4_st | A4_text | **A1A2_text** | **A2A1_text** |
| 0 | 0 | 0 | 0 | 0 | .8 | 0 |
| A2A3_text | A3A2_text | A3A4_txt | DOB_Eur | DOB_US | Telephone | Tel_local |
| .8 | 0 | 1 | .8 | .8 | - | - |

The output of the data field matching process is a vector of similarity scores indicating the similarity level of the data fields within the two records. The data field matching module is capable of doing a user-defined number and type of comparisons between two data records and generating a score for each – i.e the user will define which fields / elements of one record will be compared to which elements in the other record. The user will also define which matching algorithm is used for each comparison. In defining these parameters the user can:

- Select a default matching configuration predefined by SiVTech for a specified field type.
- Select the required matching routine for a particular data field type from a menu of predefined routines.
- Input a customised matching routine

Data Record Scoring Module 13

**Objective**

The aim of the data record scoring is to generate a single similarity score for a record pair which accurately reflects the true similarity of the record pair relative to other record pairs in the dataset. This is done by using a variety of routines to compute a similarity score from the similarity vector produced during the previous module.

**Process**

The similarity vector output from the field scoring module is input to the record scoring module. Here a set of routines are applied to derive the score.

There are two different types of routine used for this computation:

- Rule based routines – these routines use a set of rules and weights to compute an overall score from the vector. The weights are used to take into account that some fields are more indicative of overall record similarity than others. The rules are used to take into account that the relationship between individual field scores and overall score may not be linear. Following is an example of a rule based computation.

  FN = Largest of (FN_stan ,FN_Root)

  SN = Largest of (SN_stan, FN_Root )

  A1_text = Largest of (A1_text, A1A2_text)

  A2_text = Largest of (A2_text, A2A1_text, A2A3_text)

  A3_text = Largest of (A3_text, A3A2_text)

  DOB = Largest of (DOB_Eur, DOB_US)

  Score = FN + SN +A1_text+A2_text+A3_text+A4_text+(A1st+A2st+A3st+A4st)/4

- AI based routines – these routines automatically derive an optimum match score computation algorithm based on examples of correct and incorrect matches identified by the user. Depending on the situation – the type of AI technology used may be either Neural Networks or Case Based Reasoning.

The optimum routine required to derive the most accurate similarity scores for all record pairs are highly specific to the types and quality of data within a particular dataset. For this reason default routines are will generally not give the best match

accuracy. In order to achieve top levels of accuracy, a trial and error process is required to "tune" the scoring routine. This requires the user to:

- run the whole matching process a number of times for a portion of the dataset.
- inspect the results after each run to check the proportion of correct and incorrect matches.
- manually adjust the parameters of the score computation routine.

This process is extremely difficult to do with the rule based routine as there are a large number of variables to tweak. However the AI based system is ideal for this process. It removes the need to tweak different variables as the AI technology derives a new score computation routine automatically based on the learning from the manual inspection of the match results.
Since the AI process requires training data, the standard process is to use a rule based routine on the first training run and use an AI routine thereafter.

**Module Design**

The Record scoring module is designed to allow user selection or setup of both the rules based and AI based routines. The user will configure the rule based routine by:

- selecting from a menu of rule-based routine configurations predefined for common dataset types.
- selecting a predefined configuration but adjusting individual parameters (e.g. weighting of a certain field type).
- defining a customised routine.

The user will setup the AI based routine by:

- Selecting a recommended AI Algorithm type for the particular matching conditions (one-off batch matching, ongoing periodic matches etc.)
- selecting from a menu of configurations of that AI algorithm predefined for common dataset types.
- selecting a predefined configuration but adjusting individual parameters.

It will be appreciated that the system achieves:-

- fast and easy set up and configuration of new matching processes involving new datasets or match criteria.
- easy set up of adhoc matching analyses.
- scheduling of ongoing periodic matching processes using predefined configurations.
- callable from third party applications or middleware.
- capability to read data from a range of input data formats.
- range of output data functions.

Key attributes of the system are:

1. **accuracy** : capable of delivering highly accurate automated matching through the use of complex layers of processing and matching routines to compensate for the full range of data matching problems. It minimises number of true matches not identified and non-matches labelled as matches.
2. **configurability** : enables easy setup of customised routines often required due to the highly specific features of individual datasets. Allows user to select parameters based on knowledge of:
    - which fields likely to be most indicative of a match,
    - likely quality of individual fields
    - likely problems with fields / elements

Uses "wizard" type process to help user configure bespoke routines to

- remove problem characters within fields
- transform elements into standardised formats

3. **Ease of set up**: built-in intelligence to facilitate high accuracy set up and tuning by a non-expert user. Setup process leverages users knowledge of the data – but guides user on development of processing routines. Uses articial intelligence to automatically tune the matching process based on examples of good and bad matches as verified by user.
4. **Speed**: Uses smart processing to quickly reduce dataset to subset of "all possible matches" Uses highspeed pipeline to maximise processing speed.
5. **Open Architecture**: Engine architecture is uses component – based design to facilitate easy integration with other systems or embedding of core engine within other technologies

The invention is not limited to the embodiments described but may be varied in construction and detail.

Claims

1. A data quality system for matching input data across data records where the type and volume of input data is highly flexible, the system comprising:-

means for carrying out pre-processing routines on the input data to remove noise or reformat the data, and

means for matching record pairs based on measuring similarity of selected field pairs within the record.

2. A system as claimed in claim 1, wherein the matching means comprises means for extracting a similarity vector by generating a similarity score for each pair of fields in records being matched, a set of scores for a pair of records being a vector.

3. A system as claimed in claim 2, wherein the vector extraction means comprises means for executing string matching routines on pre-selected fields of the records.

4. A system as claimed in claim 2 or 3, wherein the matching means comprises record scoring means for converting the vector into a single similarity score representing overall similarity of two records.

5. A system as claimed in claim 4, wherein the record scoring means comprises means for executing rule-based routines using weights applied to fields according to the extent to which each field is indicative of record matching.

6. A system as claimed in claim 4, wherein the record scoring means comprises means for computing scores using AI (artificial intelligence) techniques to

deduce from examples given by the user the optimum algorithm for computing the score from the vector.

7. A system as claimed in claim 6, wherein the AI technology used is Cased Based Reasoning (CBR).

8. A system as claimed in claim 6, where AI technique used is Neural Nets.

9. A system as claimed in any preceding claim, wherein the pre-processing means comprises a standardisation module comprises means for transforming each data field into a number of data fields each of which is a variation of the original.

10. A system as claimed in claim 6, wherein the standardisation module comprises means for splitting a data field into multiple field elements, converting the field elements to a different format, removing noise characters, and replacing elements with equivalent elements selected from an equivalence table.

11. A system as claimed in any preceding claim, wherein the pre-processing means comprises a grouping module comprises means for grouping records according to features to ensure that all actual matches of a record are within a group.

12. A system as claimed in claim 8, wherein the grouping module comprises means for applying a key letter process for grouping.

13. A system as claimed in any preceding claim, wherein the system further comprises a configuration manager comprises means for applying configurable settings for the pre-processing means and the matching means.

14. A data quality system substantially as described with reference to the drawings.

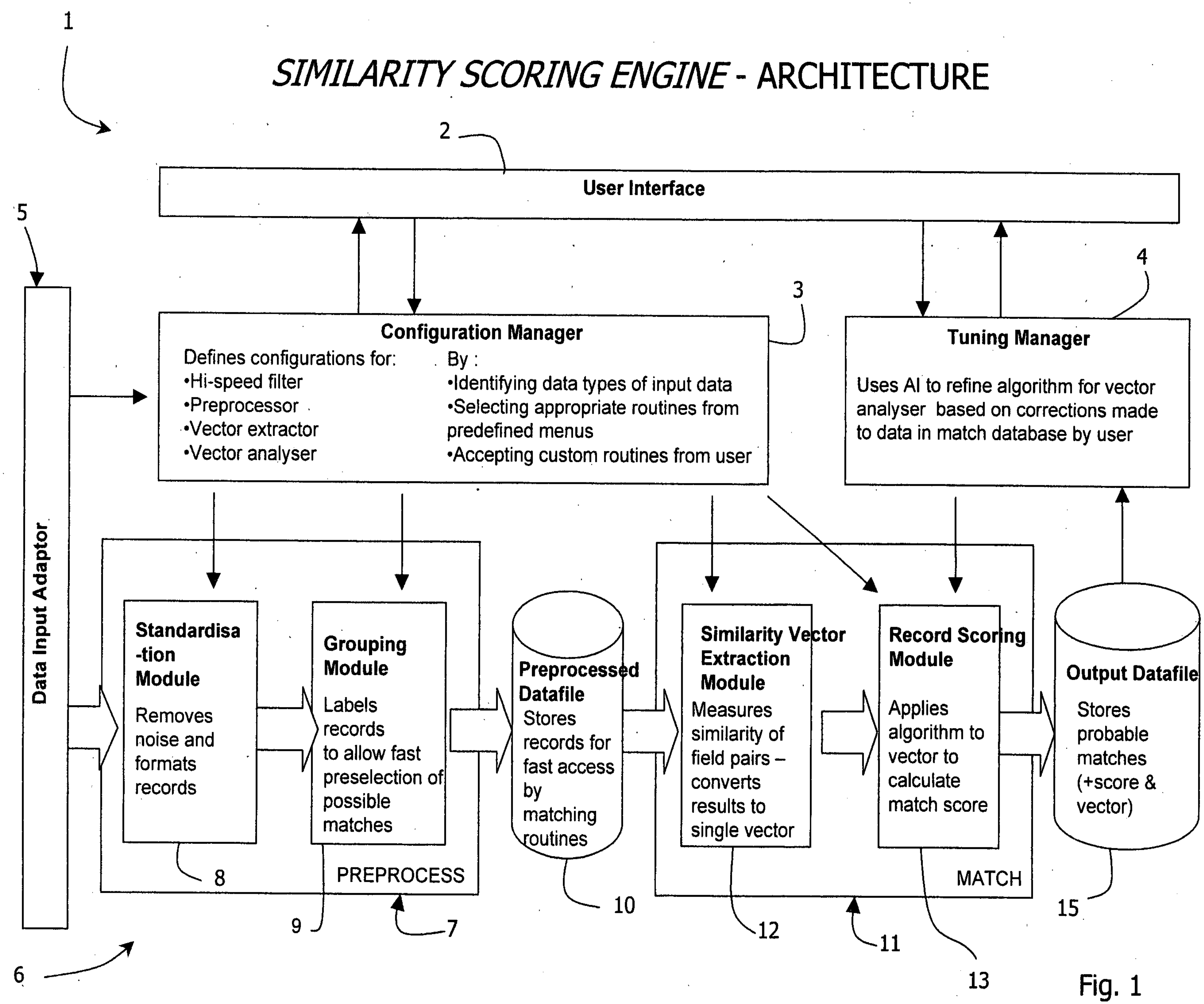
SIMILARITY SCORING ENGINE - ARCHITECTURE
User Interface
Data Input Adaptor
Configuration Manager
Defines configurations for:
•Hi-speed filter
•Preprocessor
•Vector extractor
•Vector analyser
By :
•Identifying data types of input data
•Selecting appropriate routines from predefined menus
•Accepting custom routines from user
Tuning Manager
Uses AI to refine algorithm for vector analyser based on corrections made to data in match database by user
Standardisa-tion Module
Removes noise and formats records
Grouping Module
Labels records to allow fast preselection of possible matches
PREPROCESS
Preprocessed Datafile
Stores records for fast access by matching routines
Similarity Vector Extraction Module
Measures similarity of field pairs – converts results to single vector
Record Scoring Module
Applies algorithm to vector to calculate match score
MATCH
Output Datafile
Stores probable matches (+score & vector)
1
2
3
4
5
6
7
8
9
10
11
12
13
15

Fig. 1